

[illegible]

FIG. 1

195

194

MCB Address:
0x0070030

SPR Address: 0x0030

Reset value: 0x00000000

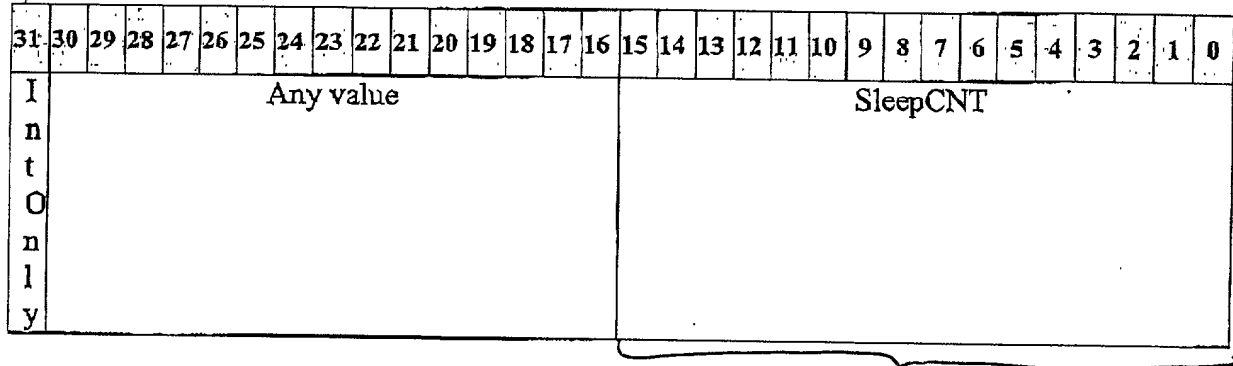


FIG. 2A 197

FIG. 2A 197

200

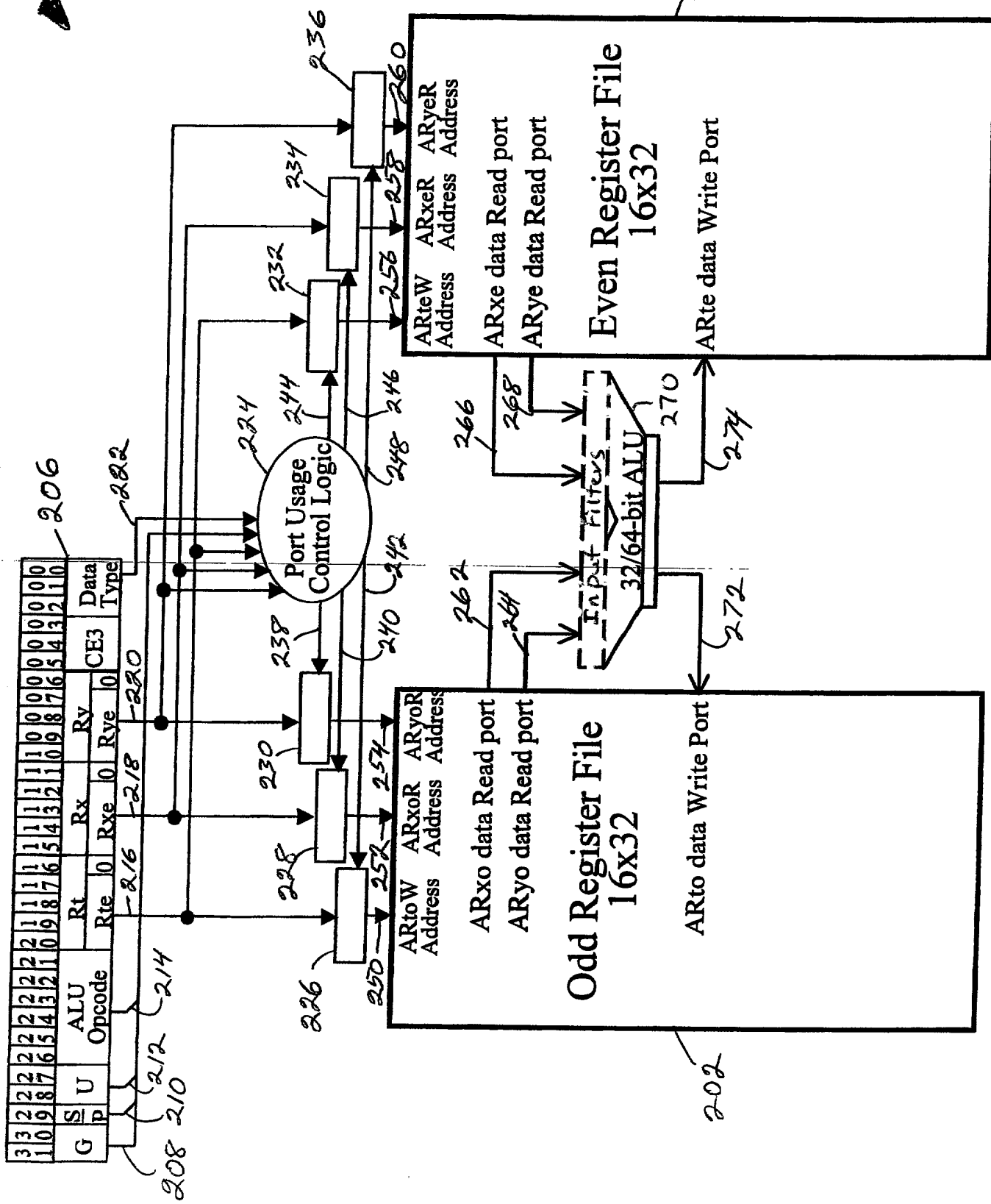


FIG. 28

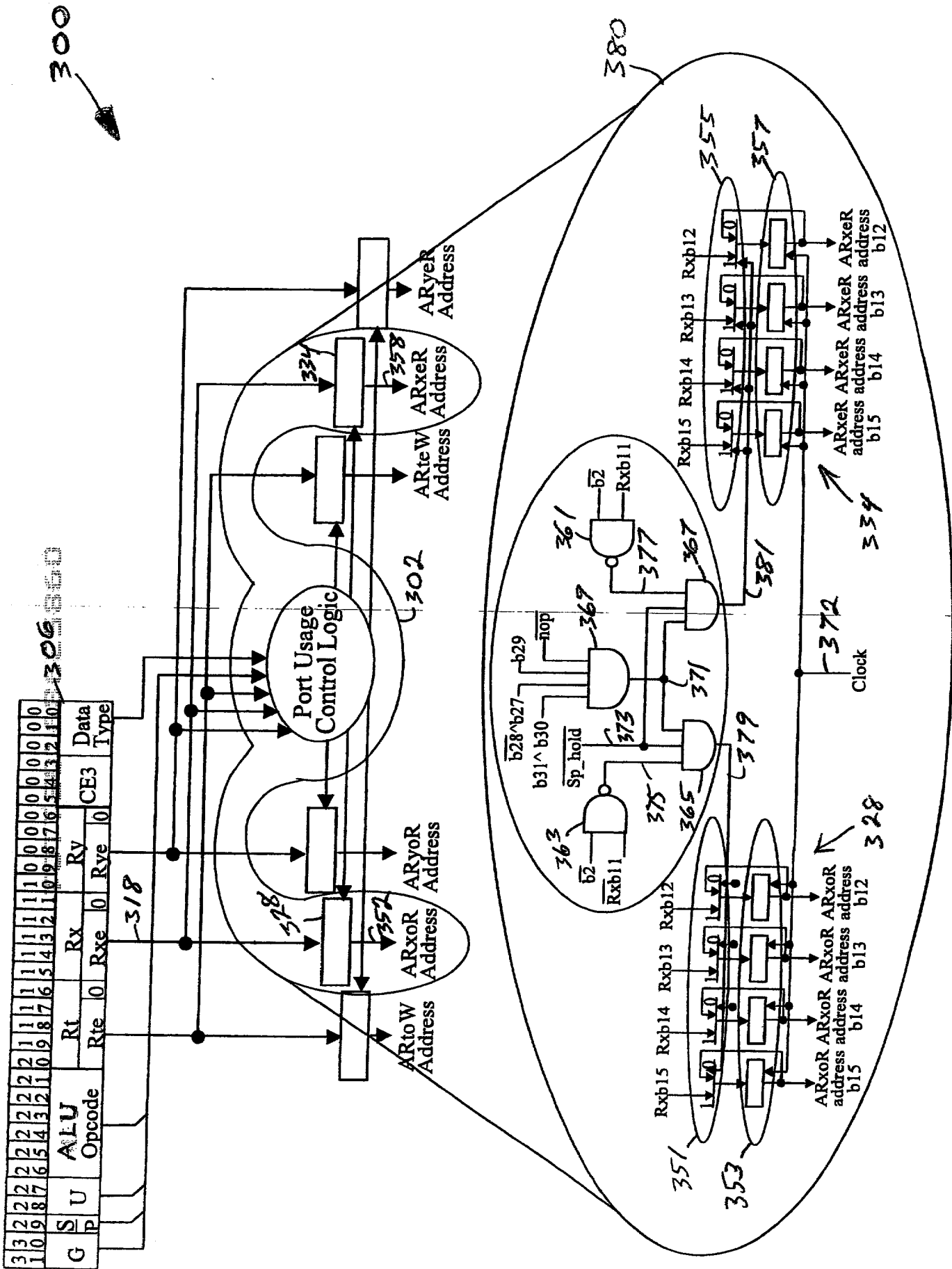


FIG. 3A

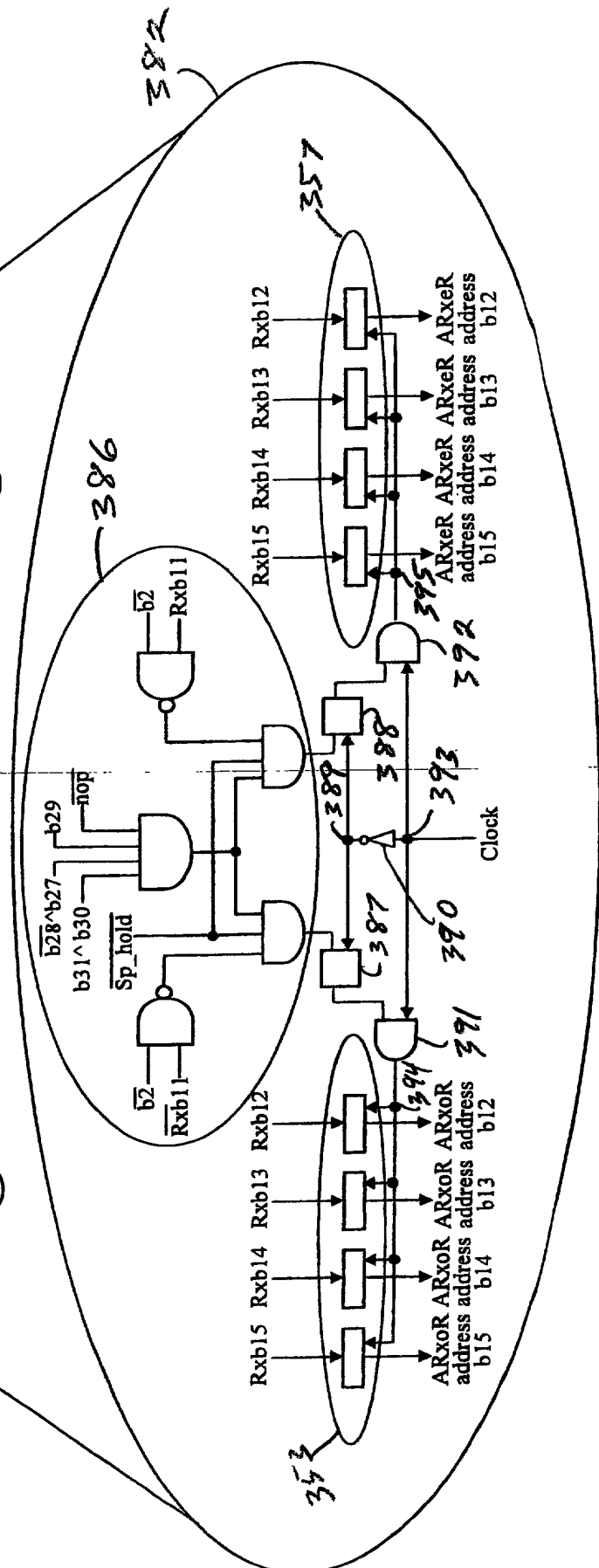
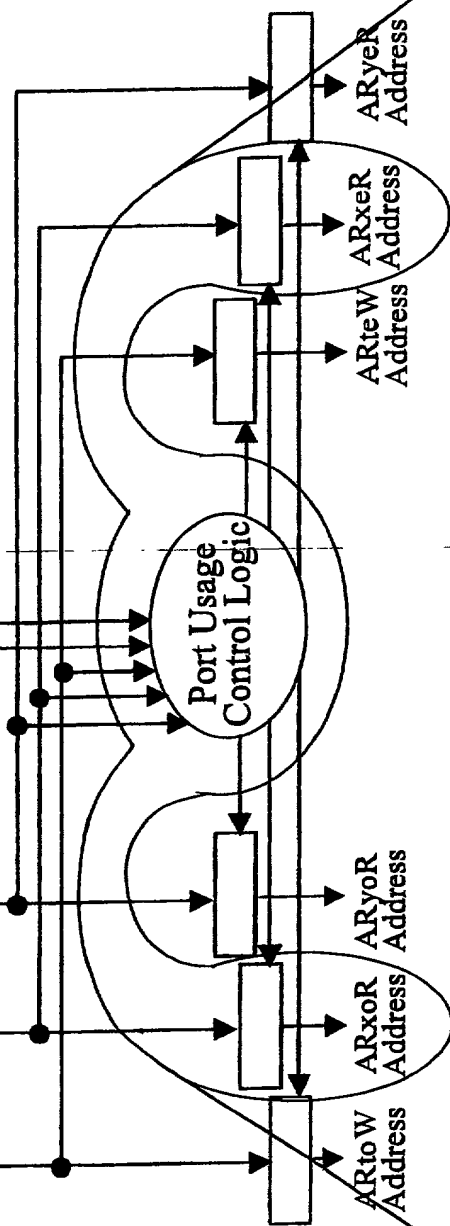
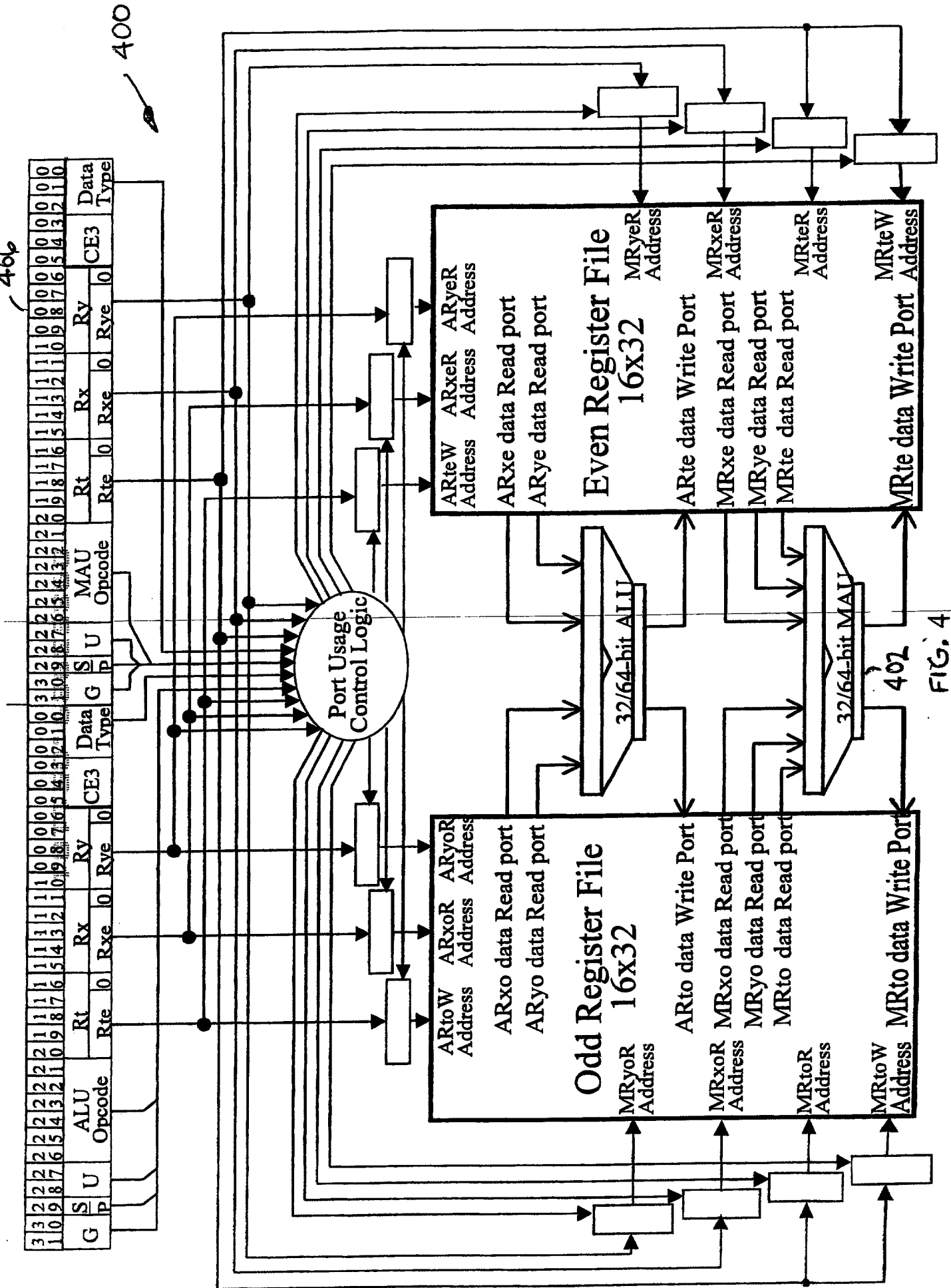
[illegible]

FIG. 3B



ALU Operation Encoding

[illegible]

Arithmetic Scalar Flags Affected (on least significant operation)

C = 1 if a carry occurs, 0 otherwise

N = MSB of result

$V = 1$ if an overflow occurs, 0 otherwise

Z = 1 if result is zero, 0 otherwise

See also ASF Definitions in chapter on Conditional Execution.

Cycles: 1

Arithmetic Execution Unit - 504

00 = ALU

01 = MAU

10 = DSU

11 = Reserved

523627

Integer Data Packing

000 = 4 Bytes (4B)

001 = 2 Halfwords (2H)

010 = 1 Word (1W)

011 = Reserved

100 = 8 Bytes (8B)

101 = 4 Halfwords (4H)

110 = 2 Words (2W)

111 = 1 Doubleword (1D)

09.5x9

FIG. 5A

Description

The sum of source registers Rx and Ry is stored in target register Rt.

Syntax/Operation

Rt, Rx, Ry ← Rx + Ry

520

Instruction	Operands	Operation	ACF
ADD.[SP][AM].1D	Rte, Rxe, Rye	Rto Rte ← Rxo Rxe + Ryo Rye	Doubleword
[TF].ADD.[SP][AM].1D	Rte, Rxe, Rye	Do operation only if T/F condition is satisfied in F0	None
ADD.[SP][AM].1W	Rt, Rx, Ry	Rt ← Rx + Ry	Word
[TF].ADD.[SP][AM].1W	Rt, Rx, Ry	Do operation only if T/F condition is satisfied in F0	None
ADD.[SP][AM].2W	Rte, Rxe, Rye	Rto ← Rxo + Ryo Rte ← Rxe + Rye	Dual Words
[TF].ADD.[SP][AM].2W	Rte, Rxe, Rye	Do operation only if T/F condition is satisfied in F0	None
ADD.[SP][AM].2H	Rt, Rx, Ry	Rt.H1 ← Rx.H1 + Ry.H1 Rt.H0 ← Rx.H0 + Ry.H0	Dual Halfwords
[TF].ADD.[SP][AM].2H	Rt, Rx, Ry	Do operation only if T/F condition is satisfied in F0	None
ADD.[SP][AM].4H	Rte, Rxe, Rye	Rto.H1 ← Rxo.H1 + Ryo.H1 Rto.H0 ← Rxo.H0 + Ryo.H0 Rte.H1 ← Rxe.H1 + Rye.H1 Rte.H0 ← Rxe.H0 + Rye.H0	Quad Halfwords
[TF].ADD.[SP][AM].4H	Rte, Rxe, Rye	Do operation only if T/F condition is satisfied in F0	None
ADD.[SP][AM].4B	Rt, Rx, Ry	Rt.B3 ← Rx.B3 + Ry.B3 Rt.B2 ← Rx.B2 + Ry.B2 Rt.B1 ← Rx.B1 + Ry.B1 Rt.B0 ← Rx.B0 + Ry.B0	Quad Bytes
[TF].ADD.[SP][AM].4B	Rt, Rx, Ry	Do operation only if T/F condition is satisfied in F0	None
ADD.[SP][AM].8B	Rte, Rxe, Rye	Rto.B3 ← Rxo.B3 + Ryo.B3 Rto.B2 ← Rxo.B2 + Ryo.B2 Rto.B1 ← Rxo.B1 + Ryo.B1 Rto.B0 ← Rxo.B0 + Ryo.B0 Rte.B3 ← Rxe.B3 + Rye.B3 Rte.B2 ← Rxe.B2 + Rye.B2 Rte.B1 ← Rxe.B1 + Rye.B1 Rte.B0 ← Rxe.B0 + Rye.B0	Octal Bytes
[TF].ADD.[SP][AM].8B	Rte, Rxe, Rye	Do operation only if T/F condition is satisfied in F0	None

FIG. 5B

MPYA - Multiply Accumulate

FIG. 6A

Encoding																															
Group	S/P	Unit	MAUpcode	Rte	0	Rx	Ry	C=3	MPack																						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax/Operation

Instruction	Operands	Operation	ACF
Word			
MPYA.[SPIM.1 SUJW	Rte, Rx, Ry	Do operation below but do not affect ACFs	None
MPYA[CNVZ]. SPIM.1 SUJW	Rte, Rx, Ry	Rto Rte \leftarrow Rto Rte + (Rx * Ry)	F0
[TF].MPYA.[SPIM.1 SUJW	Rte, Rx, Ry	Do operation only if T/F condition is satisfied in ACFs	None
Dual Halfwords			
MPYA.[SPIM.2 SUJH	Rte, Rx, Ry	Do operation below but do not affect ACFs	None
MPYA[CNVZ]. SPIM.2 SUJH	Rte, Rx, Ry	Rto \leftarrow Rto + (Rx.H1 * Ry.H1) Rte \leftarrow Rte + (Rx.H0 * Ry.H0)	F1 F0
[TF].MPYA.[SPIM.2 SUJH	Rte, Rx, Ry	Do operation only if T/F condition is satisfied in ACFs	None
Quad Bytes			
MPYA.[SPIM.4 SUBJ	Rte, Rx, Ry	Do operation below but do not affect ACFs	None
MPYA[CNVZ]. SPIM.4 SUBJ	Rte, Rx, Ry	Rto.H1 \leftarrow Rto.H1 + (Rx.B3 * Ry.B3) Rto.H0 \leftarrow Rto.H0 + (Rx.B2 * Ry.B2) Rte.H1 \leftarrow Rte.H1 + (Rx.B1 * Ry.B1) Rte.H0 \leftarrow Rte.H0 + (Rx.B0 * Ry.B0)	F3 F2 F1 F0
[TF].MPYA.[SPIM.4 SUBJ	Rte, Rx, Ry	Do operation only if T/F condition is satisfied in F0	None

FIG. 6B

Arithmetic Scalar Flags Affected (on least significant operation)

C = Not affected
N = MSB of result
V = Not affected
Z = 1 if result is zero, 0 otherwise
Cycles: 2

Arithmetic Execution Unit

00 = ALU
01 = MAU
10 = DSU
11 = Rese

Instruction Group

- 00 = Reserved
- 01 = Flow Control
- 10 = Load/Store (LU, SU)
- 11 = Arithmetic/Logical (ALU)

Mpack - Multiply Data Packing

000 = Reserved
001 = 2 Halfwords (2H)
010 = 1 Word (1W)
011 = Reserved
100 = Reserved
101 = 4 Halfwords (4H) for MPYH and MPYL
110 = Reserved
111 = Reserved

SP/PE Select
0 = SP
1 = PE

FIG. 6C

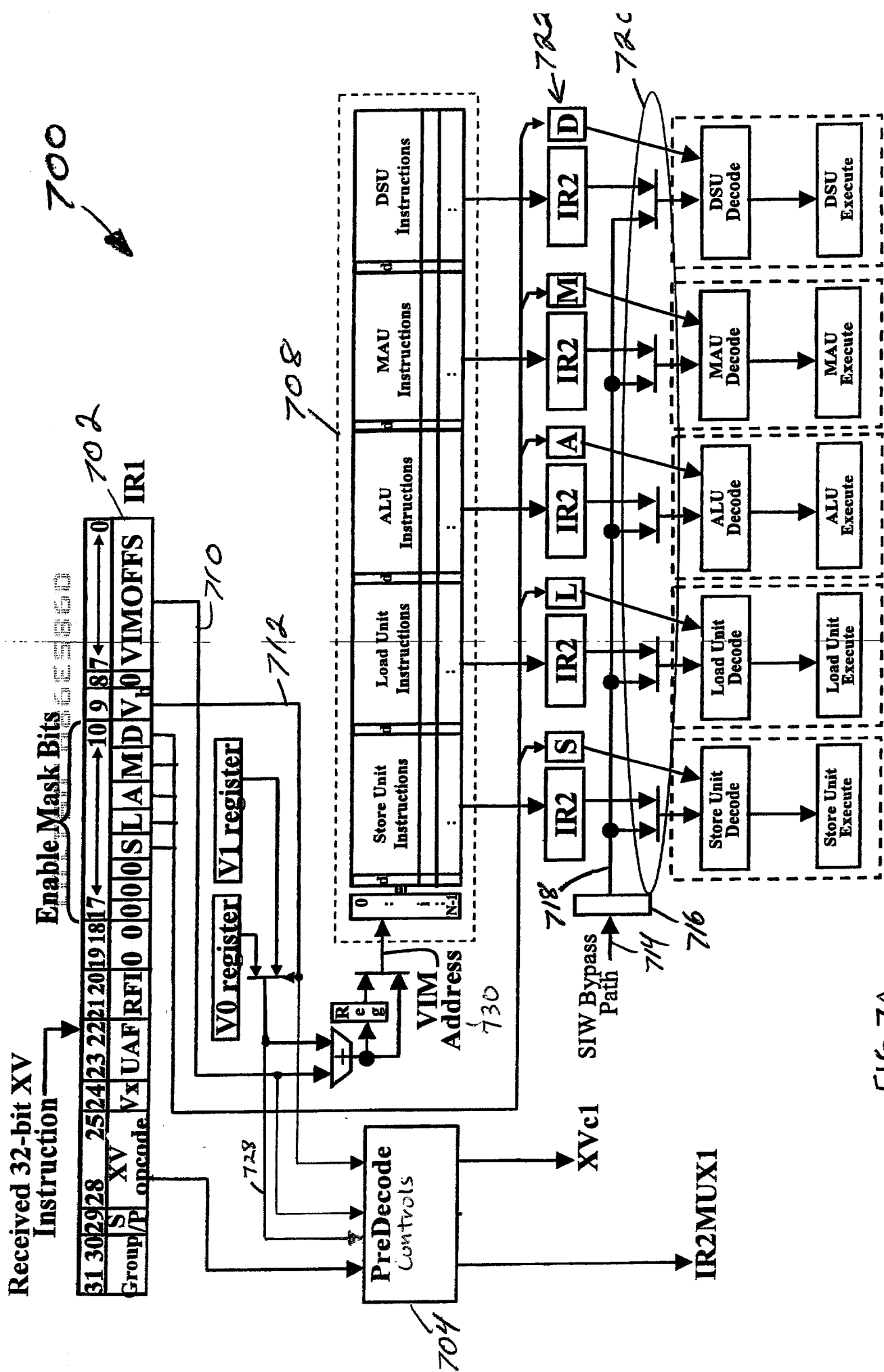


FIG. 7A

Received 32-bit XV2 Instruction

FIG. 8

800

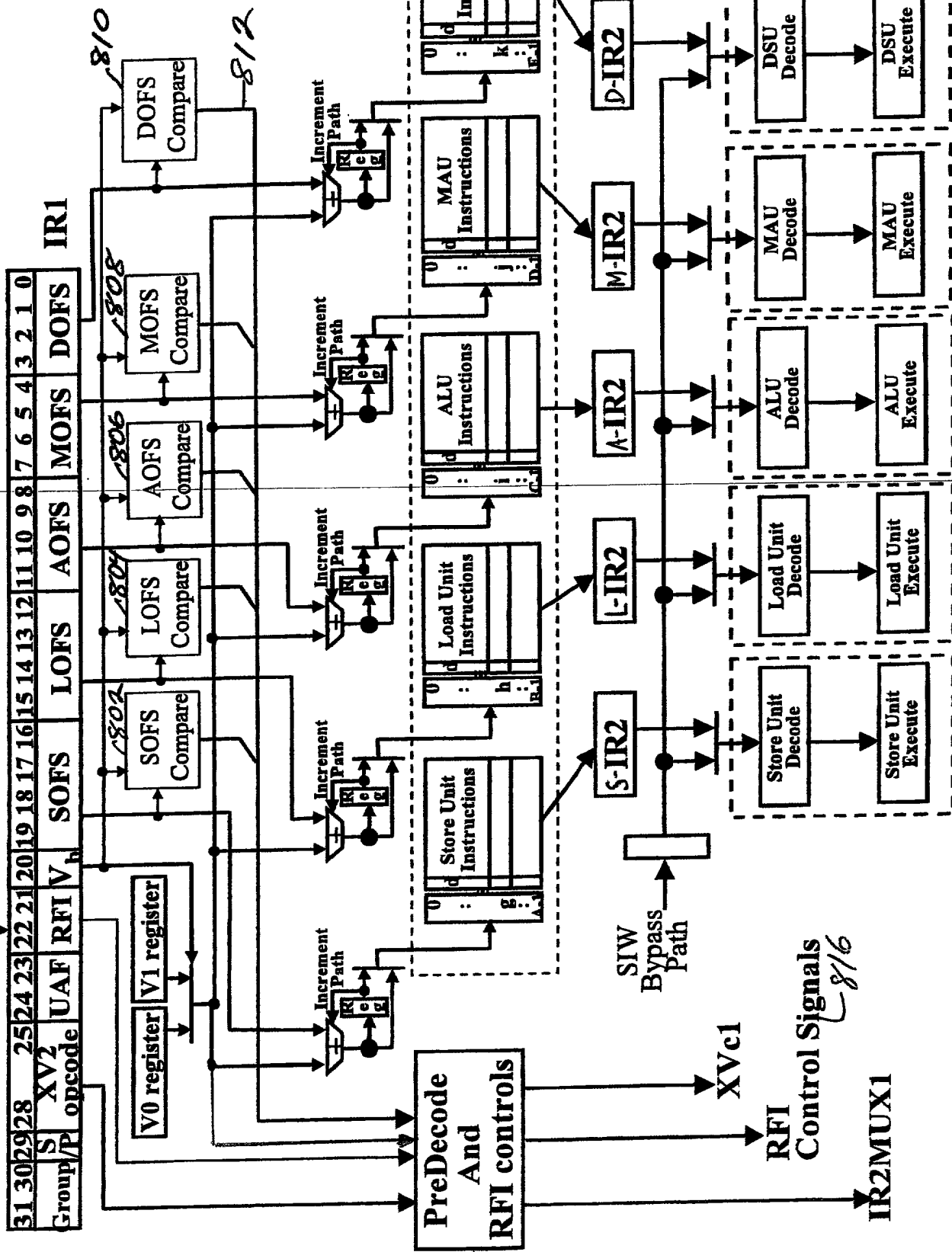


FIG. 8